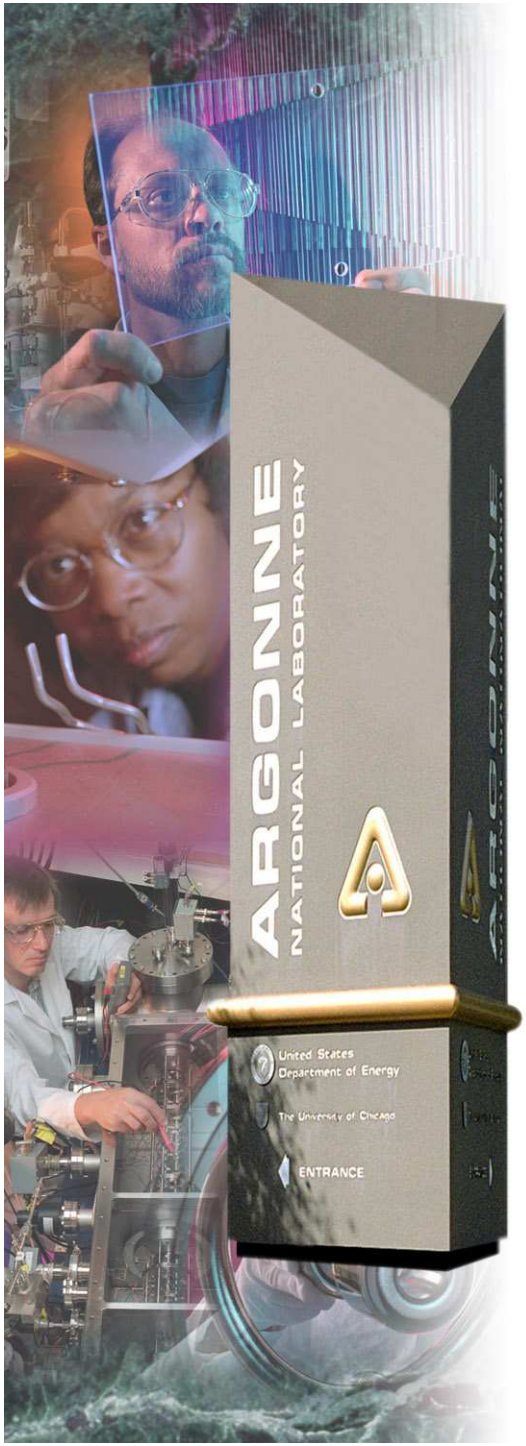


IRMIS: Future Developments

IRMIS/Relational Database Workshop
Argonne National Laboratory
D. Dohan, June 12, 2006

Argonne National Laboratory

Office of Science
U.S. Department of Energy

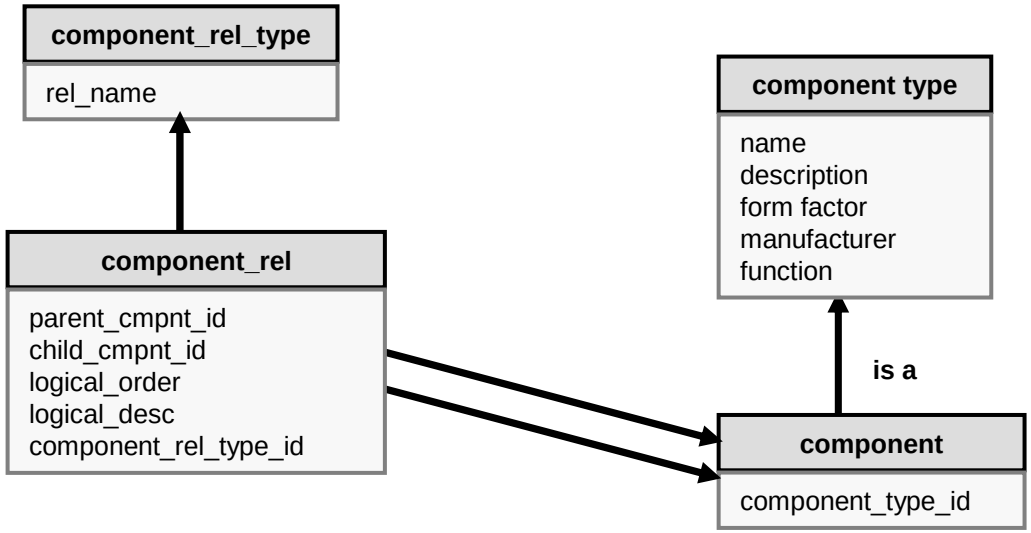
A U.S. Department of Energy
Office of Science Laboratory
Operated by The University of Chicago

Component Connections

The IRMIS schema, how components are interconnected:

- **physically (housing hierarchy)**
- **logically (control hierarchy)**
- **power grid (power hierarchy)**

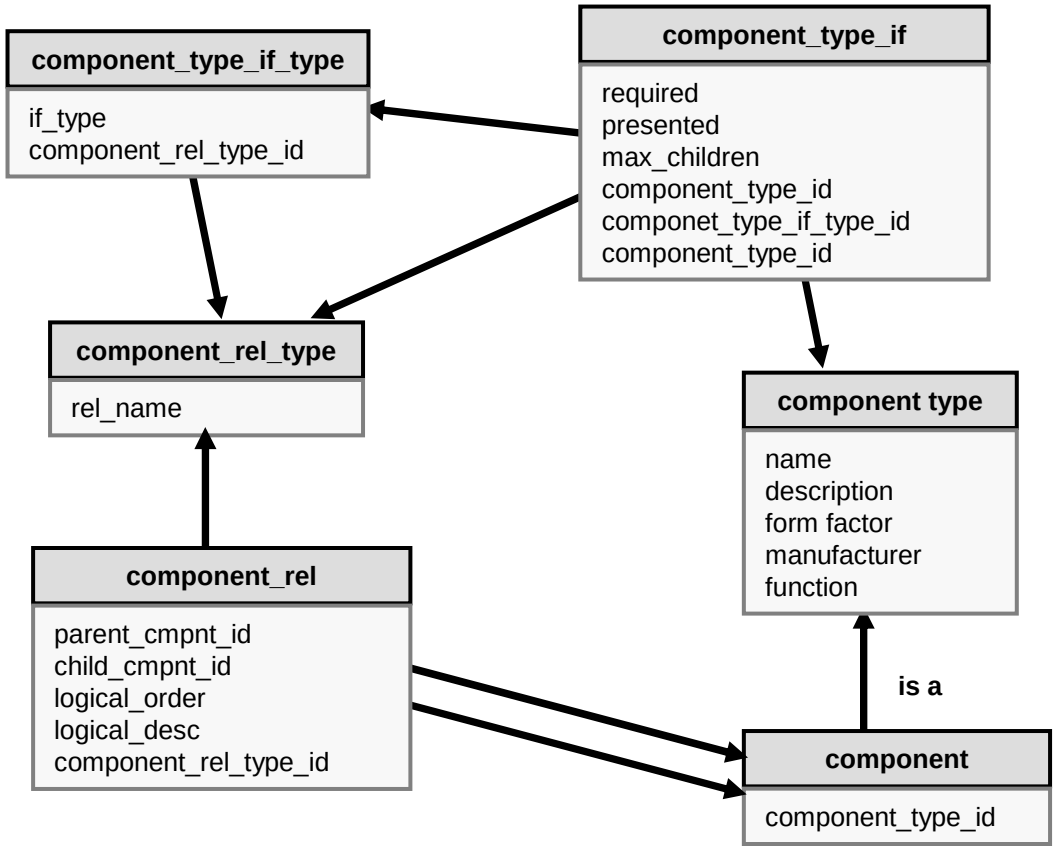
• Each relationship type is hierarchical in nature (each child has a unique parent in each hierarchy). The hierarchies are modeled as node/edge graphs (DAGs).



Component Interfaces

- **Each component type has the following properties:**
  - **a set of ‘required’ interfaces**
  - **a set of ’provided’ interfaces**
- **An ‘interface’ (both required and provided) restricts <<business rule>> the types of parent/child relationships that can take place (in any hierarchy)**
- **Thus an AllenBradley module (requires interface ‘AB_Slot’) cannot be inserted into a VME Chassis, which only provides an interface ‘VME_Slot’**

Component Interfaces - Schema

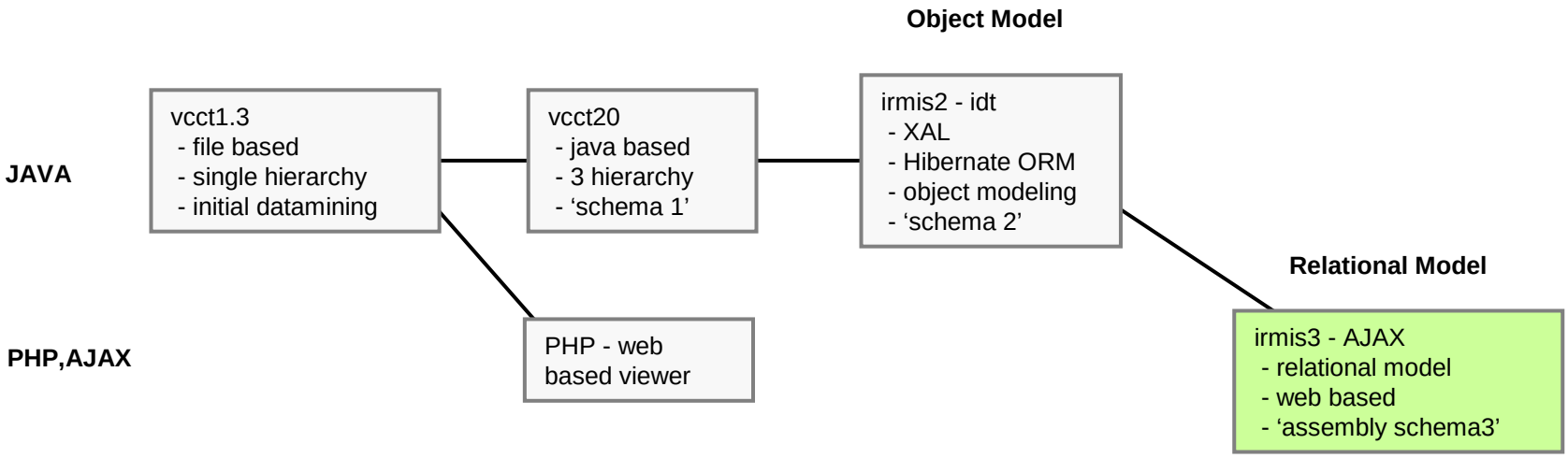


IRMIS Future Developments

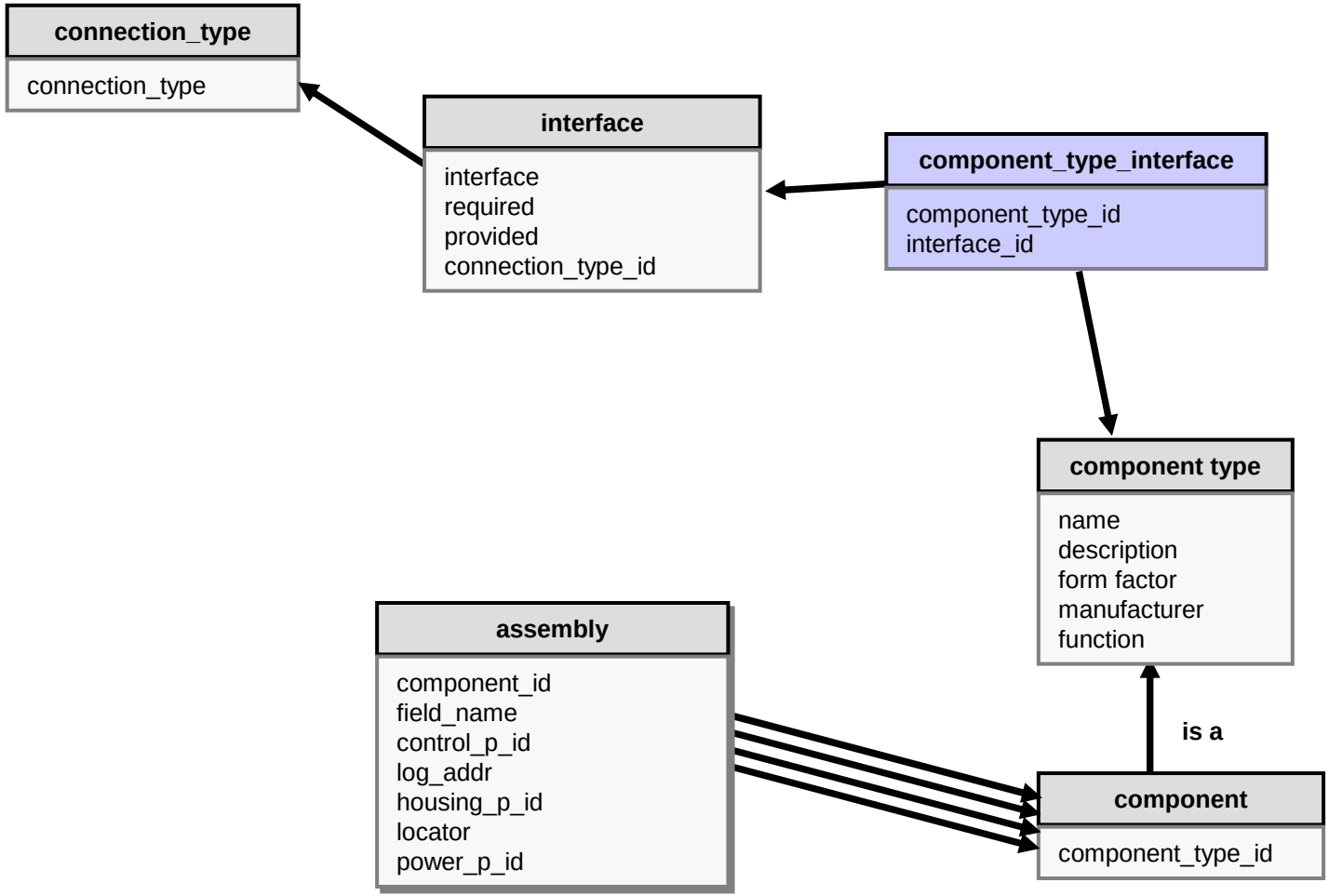
- Relating Process Variables to Components -- Data Mining
  - Query PVs for records with field_type 'INP' or 'OUT'
  - Look for a characteristic signature in the INP string. Eg AB #Lx Ay Cz Sv
  - This implies the existance of an Allen Bradley module (use DTYP for the specific module type). From IRMIS, we know which IOC has this component.
- By querying the 'control interface required' for the component, we have a limited number of component types that the component can connect to - (often only one). In the case of the A/B - we are left with the choice of an 8-slot or 12-slot chassis. Requires user input (once).
- Querying the required control interface of the chassis results in a choice of a PLC-5 or a remote adapter. The adapter number of this new component is obtained from the INP string.
- In this manner, one recursively applies the interface required/provided rule to extract the 'infrastructure' components that are implied by the existance of a given process variable.

IRMIS Future: Object vs Relational Modeling

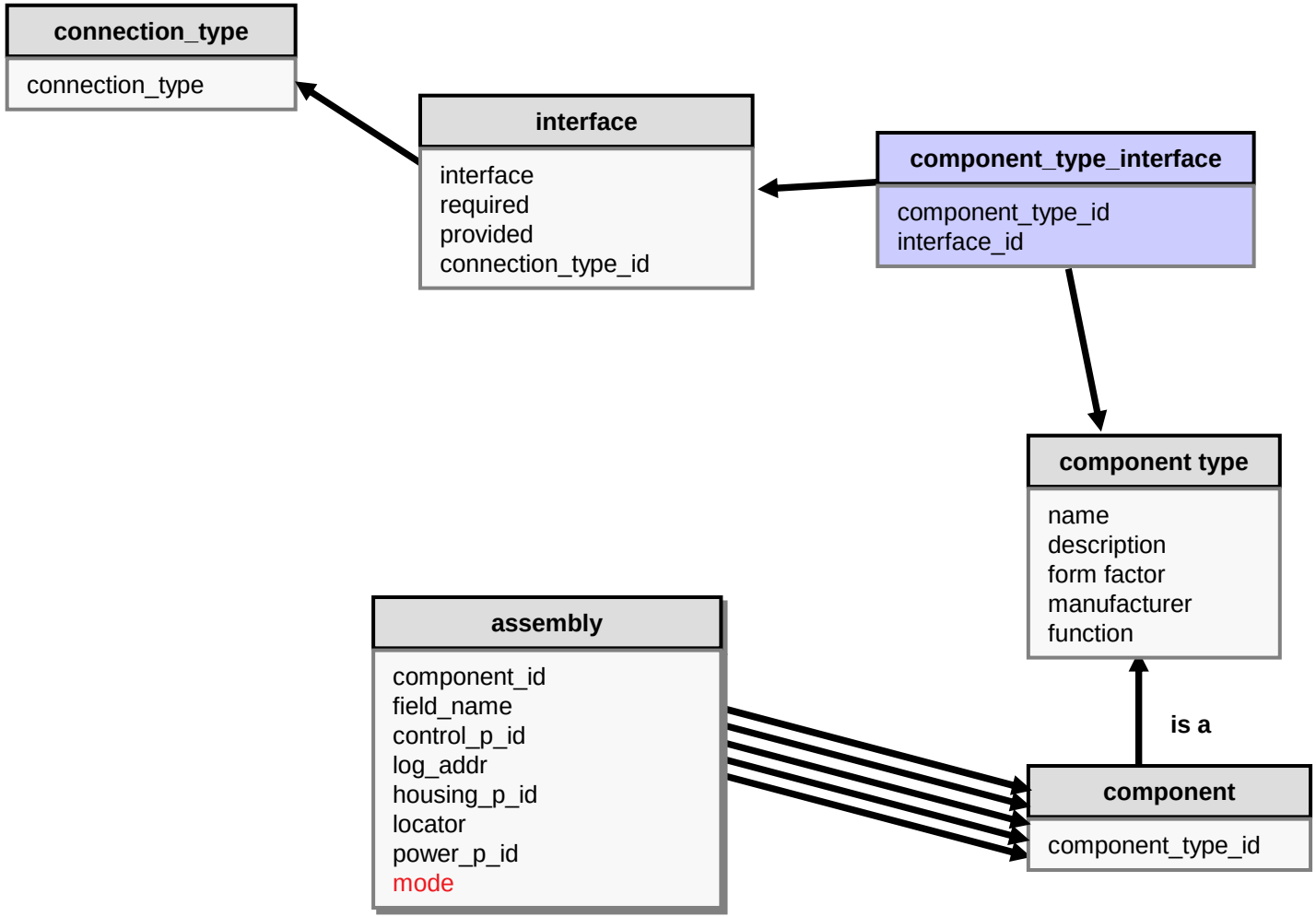
- **The data mining experience - object or relational modeling?**



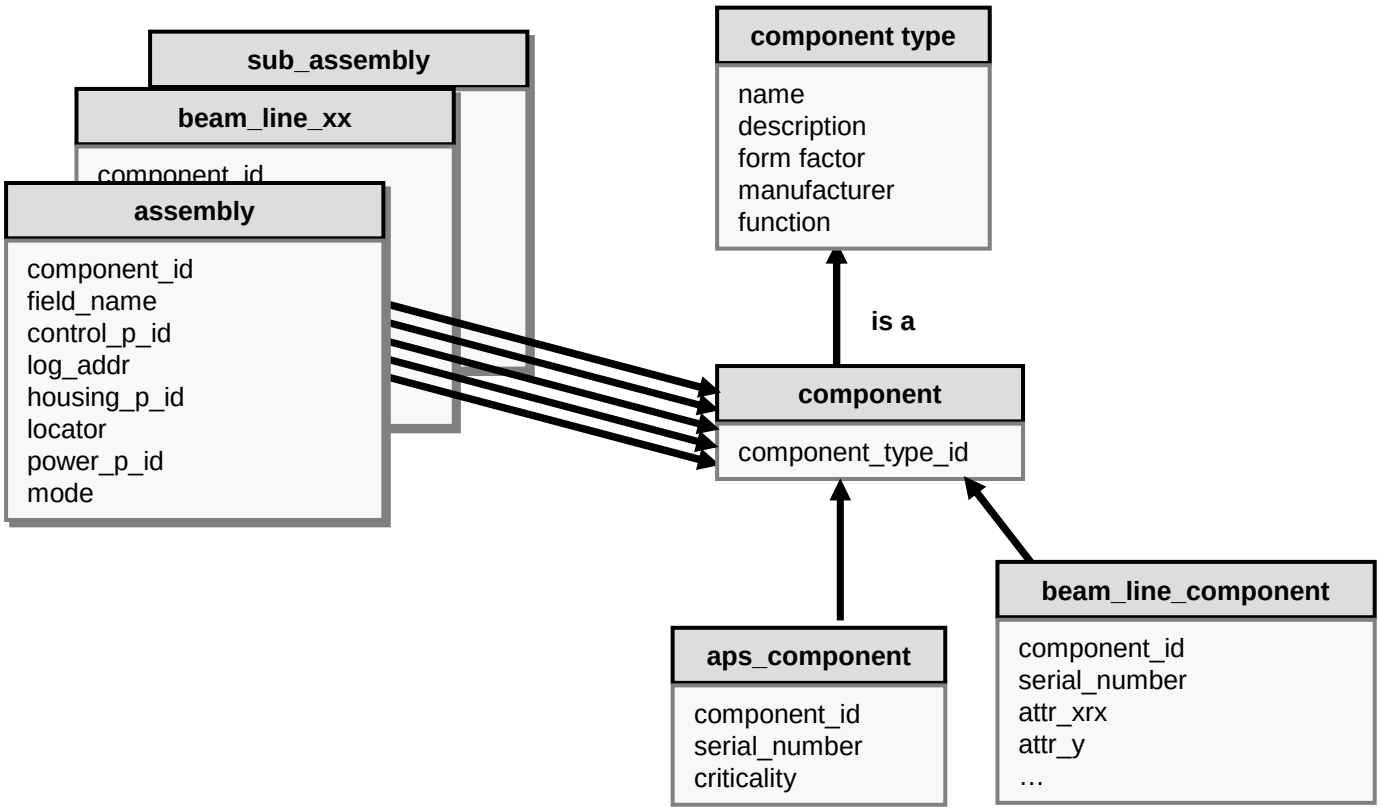
Component Interfaces - Schema



Component Interfaces - Schema



Component Interfaces - Schema



IRMIS Future Dev: Signal Tracing

- A component has in general a number of input signals, and a number of output signals.
- Each output signal is some function of the input signals, (and of the component itself).
- Extending the definition of a component output pin:
  - Each component output pin has a 'transfer function' associated with it.
  - The transfer function is similar to a line in a calc record:
    - Example: A summing component. O=(A+B)/2. Output pin O is the sum of input pins A and B.
    - Example: Fanout. $O_j = I_0$ Output in Pin J = Input on Pin 0;
    - *(the transfer functions are indicated in green)*
- With the pin transfer functions in place, one can now do end-to-end signal tracing (at least up-hill!).
- In the case of the APS timing system, it would thus be possible to trace the source of any event receiver module back to its ultimate source.

IRMIS Future Dev: Fault Tracing

- **Trace failed PVs to locate the common part of the failed field bus path**
- **Can we have 'state of health' for general field bus communications? Or is there some signature (regex?) that can be defined to locates state of health PVs?**
  - **Auto generate alarm handler config files with these PVs on IOC reboot**
- **Fault tracing is helped considerably if we can do general signal tracing.**

IRMIS Future Dev: Component Transfer Function